

Implementatieplan

Practicum onderdeel #1: Imageshell and Intensity



Jelle van den Broek en Mart Notermans

Vision

V2A

27-03-2020

Inhoud

Doel	2
Methoden	2
Keuze	3
Implementatie	3
Evaluatie	3
Bronvermelding	4

Doel

Het doel van deze implementatie is om een efficiëntere manier te gebruiken om RGB afbeeldingen om te zetten naar intensity afbeeldingen (grijswaarden). Hierdoor zal het face recognition programma sneller en efficiënter moeten werken.

Hiervoor gaan wij een nieuwe image shell voor de RGB image en voor de Intensity image schrijven. De nieuwe image shell moet efficiënt werken zodat de conversie snel kan worden uitgevoerd en de gezichtsherkenning nog steeds accuraat is.

Methoden

De paper 'Color-to-Grayscale: Does the Method Matter in Image Recognition' van Christopher Kanan en Garrison W. Cottrell ^[1], heeft 13 verschillende manieren vergeleken om een RGB afbeelding te converteren naar een afbeelding met grijswaarden.

De eerste en meest simpele methode genoemd in deze paper is de afbeelding te converteren naar een intensity image door de gemiddelde waarde te berekenen. De volgende formule wordt gebruikt:

$$G_{Intensity} \leftarrow \frac{1}{3}(R + G + B)$$

Een andere manier van een afbeelding converteren naar grijswaarden is luminance, ontworpen om overeen te komen met de menselijke helderheidsbeleving. Dit gebeurt door een gewogen combinatie van de RGB kanalen te gebruiken. Luminance wordt gebruikt in MATLAB's 'rgb2gray' function ^[2] en wordt veel gebruikt in computer vision.

$$G_{Luminance} \leftarrow 0.3R + 0.59G + 0.11B$$

Luma is een vergelijkbare manier die vaak gebruikt wordt in high-definition televisies (HDTV).

$$G_{Luma} \leftarrow 0.2126R' + 0.7152G' + 0.0722B'$$

Luma maakt gebruik van standaard gamma correctie, genoteerd in bovenstaande formule met een X' . De formule voor deze standaard correctie is als volgt ^{[1][3]}.

$$Gamma_{correctie} = t' = t^{1/2.2}$$

Keuze

Om de meest efficiënte manier van RGB naar grijswaarden conversie te bepalen, worden alle drie de methoden genoemd in het hoofdstuk 'Methoden' getest. Behalve de formules is de code gebruikt bij het testen van de drie methoden identiek. Op deze manier hoeven wij ons geen zorgen te maken over de invloed van externe variabelen. Wij hebben besloten twee verschillende datasets te gebruiken om beter te kunnen zien of er een verschil was tussen de methoden.

Implementatie

Om de image shell te maken gaan wij de gegeven functies uit 'IntensityImageStudent.h' en 'RGBImageStudent.h' aanvullen en implementeren. De formules uit de methoden worden in het bestand 'StudentPreProcessing.cpp' geïmplementeerd. Zoals eerder genoemd wordt bij het testen geen code veranderd, behalve de formules zelf. Dit doen wij om de meest betrouwbare testresultaten te behalen.

Evaluatie

De drie methoden testen wij op twee verschillende datasets. De eerste dataset bestaat uit een reeks afbeeldingen van bekende personen en de tweede dataset bestaat uit een reeks pasfoto's.

De eerste drie metingen zijn uitgevoerd door de drie methoden (Intensity, Luminance en Luma) toe te passen op de dataset met afbeeldingen van bekende personen. Daarna is de default methode, die meekwam met de voorbeeldcode, ook getest op de bekende personen dataset. Vervolgens zijn wij doorgegaan naar de dataset met pasfoto's. Ook hier zijn eerst de drie hiervoor genoemde methoden getest met daarna de default methode.

Er zijn twee bestanden met meetresultaten gemaakt, een voor elke dataset. en een meetrapport waarin deze resultaten worden besproken.

Bronvermelding

- [1] Kanan, C., & Cottrell, G. W. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition? *PLoS ONE*, 7(1). <https://doi.org/10.1371/journal.pone.0029740>

- [2] Hritik Raj. (2018, 25 juni). *MATLAB / RGB image to grayscale image conversion*. Geraadpleegd van <https://www.geeksforgeeks.org/matlab-rgb-image-to-grayscale-image-conversion/>

- [3] Wikipedia contributors. (2020, 12 januari). *Gamma correction*. Geraadpleegd van https://en.wikipedia.org/wiki/Gamma_correction